1.1 تعلم الأسمبلي باللغه العربيه - الدرس

الرئيسة | الدرس التالي | منتدى الأسمبلي | العودة الى الوحدة الأولى
| آخر تحديث : 2001/7/19 | عدد الزوار

ماهي الأسمبلي :

في قديم الزمان أيام بدايات الكمبيوتر كانت برمجة الكمبيوتر تتم بواسطة **لغة الأله Machine Language** أختصاراً **ML** *(لغة الآله هي اللغه التي تفهمها الآله مباشرة دون الحاجة الى تفسير وهي تخزن بصورة ثنائية [تركيبة من الأصفار والوحايد] في الذاكرة على شكل تعليمات ووسائط تأخذ كل واحد منها عادة مقدار 8بت=1 بايت)* وكان هذا النوع من البرمجة صعب جداً عندها طور المبرمجون أول لغة برمجة وهذه اللغه فكرتها بسيطة جداً حيث أنه بدل أن تكتب رموز الآله يتم كتابة كلمات مختصره تدل على نوع العمليه مثال (MOV,ADD,CMP) ثم ببرنامج بسيط يتم تحويل هذه الشفرة الى لغة الآله بأستخدام تخطيط واحد-الى-واحد أي أن كل سطر أو عبارة في الأسمبلي تحول الى تعليمة واحدة مقابله في لغة الآله (مثال بدل كتابة 0110000000000101 يتم كتابة mov al,5) يعرف البرنامج الذي يقوم بعملية التحويل **بالأسمبلر Assembler** ، علماً بأن هناك عدة أنواع من الأسمبلر كل نوع يختص بتقنية معينة وبعائلة معينه من المعالجات

ونحن هنا بصدد تعلم البرمجة بالأسمبلي للمعالجات المبنية على تقنية **IBM-PC** والمنتجة من شركة أنتل وهي العائلة **80×86** ويرمز لها أختصاراً **X86** وهي تضم : (8086 / 8088 / 80186 / 80286) للمعالجات 16 بت و (80386 / 80486 / 80586=بنتنيوم1 / 80686=بنتنيوم2 / 80786=بنتنيوم3 / 80886=بنتنيوم4) للمعالجات 32 بت وسوف أتطرق في دروس متقدمة الى المعالج أتنيوم 64 بت المبني بتقنية جديده كلياً لمن يرغب بمعرفة مسبقة لهذا المعالج الجديد كذلك سوف أتطرق بأذن الله الى **الكروس أسمبلر** وهي مجموعة برامج خاصة مصممه للتحويل من لغة أسمبلي لعائلة معالجات معينة الى عائله أخرى .

تعريف لغة الأسمبلي
الأسمبلي هي لغة برمجة تتكون من سلسلة من التعليمات المتتابعة كل تعليمة فيها تحول الى تعليمة مقابلة بلغة الآله .

تعريف الأسمبلر
الأسمبلر هو برنامج يقوم بتحويل التعليمات المكتوبة بلأسمبلي الى لغة الآله .

لماذا أريد أستخدام الأسمبلي :

بتعلمك لغة الأسمبلي فأنك تكشف النقاب عن الأسرار المختفيه وراء الكمبيوتر وتصبح قادراً على الفهم تماماً كيف يعمل المعالج وكيف يعمل البرنامج وبذلك تزيد خبرتك كمبرمج وبالطبع فأن الأسمبلي أقوى من اللغات العالية المستوى في التعامل مع العتاد وتعطيك مرونة عالية وقدرة وصول الى أشياء لم تكن تستطيع الوصول اليها من قبل ، كذلك هناك نوعيات من البرامج لايمكن الا برمجتها بالأسمبلي مثل الدرايفات(سواقات) الأجهزه ، كذلك فأن الأسمبلي يعطيك برامج سريعة جداً ، وبالطبع فأن بناء برنامج متطور بالأسمبلي أشبه بحفر حفرة بواسطة الملعقه فالبرغم أنك تحفر الا أنك أنت تجيد الحفر قليلاً ولكن من المحبذ جداً برمجة بعض الدوال و الأجزاء من البرامج بالأسمبلي وبقية البرنامج بواسطة لغة عالية المستوى مثل السي++.

العلاقة بين الأسمبلي واللغات الأخرى :

تعتبر كل من لغة الآله و الأسمبلي لغتين منخفضتي المستوى **Low-Level Language** أختصاراً **LLL** لأنها تكتب تعليمه تعليمه (بعض الناس يعتبر لغة السي لغة منخفضة المستوى ، وهذا الكلام أرجح الى الخطأ منه الى الصوا ب) بينما تعتبر باقي اللغات **High-Level Language** أختصاراً **HLL** وفي هذه اللغات تختفي تقنية تخطيط واحد-الى-واحد وتفسر التعليمة الواحدة الى عدد كبير من تعيلمات لغة الآله

مخطط يوضح العلاقة بين لغة الأسمبلي ولغة الآله ولغة عالية المستوى ولغة الآله

تطبيقات لغة الأسمبلي :

تتطلب كتابة البرامج بلغة الأسمبلي معرفة بالعتاد وعناية خاصة مع الأهتمام بأد ق وأقل التفاصيل ، في أيام البرمجة القديمة كان المبرمجون يكتبون برامجهم بلغة الأسمبلي لأن ذاكرة الرام وقتها كانت صغيرة (أقل من 64كيلوبايت) وهم بحاجة الى برامج أصغر وأسرع خصوصاً أ ن معالجتهم أيضاً كانت بطيئة ، مع تطور الحاسوب وتوسع سعة ذاكرة الرام وزيادة سرعته أصبحت البرامج أكثر طولاً وتعقيداً ، هذا التعقيد أدى الى أستخدام اللغات البرمجية عالية المستوى HLL مثل السي والكوبول والبيسك والباسكال والفورترن ، مرة أخرى تطور الحاسوب فأدى الى أستخدام اللغات العليا الموجهة الهدف OOP مثل السي++ والجافا والتي مكنت من كتابة برامج قوامها آلاف الأسطر والتعليمات المعقده والمتداخله .

من الصعب أ ن تلاقي برامج كبيرة مكتوبة كاملة بلغة الأسمبلي لأن كتابتها صعب والأهم من ذلك تطويرها وصيانتها أصعب بكثير ، بدل ذلك يقوم المبرمجين ببرمجة مقاطع مبرمجة برمجة مثلى بلغة الأسمبلي لأستخدمها في تنفيذ أسرع أو الوصول الى العتاد عن طريقها وباقي البرنامج بواسطة لغة عالية المستوى .

يفض المبرمجين لغة السي++ كلغة قياسية للبرمجة بلغة عليا لأن لها قدرة عالية وقوية جداً وموجهة الهدف مع القدرة على كتابة مقاطع السي فيها وهي لغة أقل أنخفاضاً وأكثر مرونة مع أستخدام الأسمبلي كعنصر مهم في الوصول الى العتاد وبرمجة الجزئيات المحتاجة للسرعة .

لا يستخدم المبرمجون شفرة الأسمبلي وسط شفرة لغة عالية المستوى عادة وأنما يستخدمونها عن طريق واجه على شكل دالة أو كائن وتحتوي هذه الدالة أو هذا الكائن على شفرة الأسمبلي المطلوبة ، وقد تستخدم روتيناً فرعياً أو دالة في لغة عالية

المستوى وأنت لاتعلم بأنك باستدعاء هذه الدالة أو الروتين الفرعي قد أستدعيت شفرة مكتوبة بلغة الأسمبلي .

لغة الآله :

لغة الآله كما ذكرنا سابقاً هي اللغة التي تسطيع الآله أو المعالج التعامل معها مباشرة ، في العائلة X86 كل معالج يحتوي يستطيع تنفيذ تعليمات المعالج الذي قبله ويملك مجموعة تعليمات موسعة وأ ضافية لا تستطيع المعالجات التي قبله تنفيذها وأما المعالج الذي يأتي بعد هذا المعالج فأنه يدعم التعليمات الموسعة للمعالج الذي قبله بينما يحتوي هو أيضاً على تعليمات جديدة وموسعة ، بأختصار اذا صممت برنامج لمعالج ما فأن المعالجات ماقبل هذا المعالج لن تستطيع تشغيله بينما المعالج نفسة والمعالجات التي بعده (من نفس العائلة) تستطيع تشغيل البرنامج .

لقد حافظت شركة أنتل على التوافقية في العائلة X86 أبتداء من المعالج 8086 وصولاً الى بانتنيوم 4=80886 ولكن الحفاظ على التوافقية يفرض قيود على تصميم المعالج وأستخدام تقنيات قديمة ، ومؤخراً قرر ت شركة أنتل ايقاف عائلة المعالج X86 عند بانتنيوم4 وقامت بأنشاء معالج جديد (غير متوافق مع العائلة X86) مبني على تقنية لا64 بت وهو المعالج اتانيوم الجديد .

مثال على لغة الآله :

التعليمية 1011000000000101 هي تعليمة بلغة الآله ومعناها أنقل الرقم 5 الى المسجل ah يقابلها بلغة الأسمبلي mov ah,5 لا 8 بت الأولى من التعليمة تشكل **شفرة التعليمة operation code**=OP-code وهي تعني نقل قيمة بطول 8 بت الى المسجل AL ، لجزء الثاني من لا 16 بتلا 8بت الأخرى تشكل الرقم 5 ثنائياً

إضغط هنا للأنتقال الى الدر سـ التالي

إعداد : أحمد الزياني

قد تتسائل ما علاقة تمثيل البيانات والعد الثنائي بالأسمبلي ؟ حسناً كما وضحت من قبل فأن الأسمبلي هي لغة قريبة جداً من لغة الآله وهي لغه منخفضة المستوى تتعامل مع العتاد والمعالج بصورة مباشرة ولكي نحقق فهماً أوسع لهذه اللغه يجب أ ن نفهم بعض الأشياء المهمة جداً في بنية المعالج .

العد الثنائي :

يتم تمثيل الشفرا ت والبيانات في ذاكرة الكمبيوتر كتواليف من الشحنات الكهربائية تأخذ قيمتين الأولى وهي وجود الشحنة ويرمز لها ب ON أو صحيح TRUE أو '1' والأخرى وهي غياب الشحنه ويركز لها ب OFF أو خطأ FALSE أو '0' ، ووجود الشحنة يكون عادة بين 4.5 الى 5.5 فولت (المعالجات الحديثه بين 2.5 الى 3.5 فولت) وغياب الشحنة يكون بين +0.5 فولت و -0.5فولت .

وحدا ت الذاكره الأساسيه في الذاكره والوحدا ت التي سنتعامل معها كثيراً هي :

أسم الوحدة		الأحتمالات	مايعادلها
البت	Bit	أحتمالين 0 و 1	1بت
البايت	Byte	256 أحتمال	8بت
الكلمة	Word	65536 أحتمال	16بت=2بايت
الكلمة المضاعفة	DoubleWord=Dword	4,294,967,296 أحتمال	32بت=4بايت=2كلمة
الكلمة الرابعية	Quafword=Qword	18,446,744,073,709,551,616 أحتمال	64بت=8بايت=4كلمات=2كلمة مضاعفة

بالطبع هذه الوحدا ت الأساسية والصغيره أما مضاعفاتها فهي :

الوحده	الرمز	مايعادلها
الكيلوبايت	KB	1024 بايت
الميجابايت	MB	1024 كيلوبايت
الجيجابايت	GB	1024 ميجابايت
التيرابايت	TB	1024 جيجابايت
البيتابايت	PB	1024 تيرابايت
الأكسابايت	EB	1024 بيتابايت
الزيتابايت	ZB	1024 أكسابايت
اليوتابايت	YB	1024 يوتابايت

تخزين الأرقام بدو ن أشارة يأخذ النطاق الآتي لكل وحده من الوحدا ت الأساسية بأستخدام طريقة بدو ن الأشارة :unsigned

أسم الوحدة	المجال	
	من	الى
البايت	0	256
الكلمة	0	65536
الكلمة	0	4,294,967,296

المضاعفة		
الكلمة الرباعية	0	18,446,744,073,709,551,616

الأسكي كود ASCII:

يتم في الحاسوب وبقية توحيد أستخدام الرموز أستخدام شفرة الآسكي كود (حالياً يعمل على تبني شفرةلا unicode وهي تسمح بتعدد اللغات في مستند واحد حيث يتم تمثيل كل حرف بأستخدام كلمة واحده=2بايت) كلمة ASCII هي أختصار ل :
American National Standard Code For Information Interchange
ويتم أستخدام هذا الكود الموحد لتسهيل تناقل البيانات ويمثل كل رمز فيه بعدد ثنائي بطول 1بايت=8بت=256أحتمال .

طريقة كتابة الأرقام في الأسمبلر :

لكتابة عدد ثنائي يوضع في آخر الرمز (b) لدلالة على أنه بايني مثال :
Binary=11010010B أما العدد العشري فلايحتاج الى أضافة وأما العدد لأساس 8 فيكتب مع المرمز (Q) في نهايته Octal=1276Q أو الرمز ((O في نهايته Octal=1276O أما العدد السداسي عشر فيكتب بوضع H في نهايته hexadecimal=0AB9CDH مع مراعاة وضع 0 اذا كان العدد يبدأ بحرف كما المثال .

يجب أن تعرف الفرق بين تخزين الرقم كرقم أو تخزينه كنص فتخزين الرقم 201 مثلاً كرقم سسيأخذ بايت واحد وهو جاهز للقيام بعمليات رياضية ومنطقيه عليه أما تخزينه كنص فسيأخذ ثلاثة بايت في البايت الأول سيخزن الرقم الخاص بالأسكي كود للرمز '2' وكما قلت يخزن كرقم يدل على الرمز أما البايت الثاني فسيخزن رقم الآسكي كود للرمز '0' أما البايت الثالث فيأخذ القيمه الخاصة بالرمز '1' في الآسكي كود أي أن الرقم خزن بطريقة "102" وليس 102 وهذه الطريقة ليست جاهز للجمع أو الطرح ولكنها ممتاز للطباعه على الشاشه ويمكن تحويل النص الى رقم والعكس .

الأعداد ذي الأشاره :

يتم تخزين الأعداد ذي الأشاره كالتالي :
العدد موجب أذا كانت البت الأخيره صفر وقيمة الرقم هي باقي البتات أي لو أخذنا رقماً من بايت واحد فأن البت رقم 7 (الثامنه - الترقيم يبدأ من الصفر) يجب أن تكون صفراً ليكون العدد موجب أما البتات من 0 الى 6 (السبعه الأولى) فتشكل قيمة الرقم أما أذا كان العدد سالب فأن البت الأخيرة تساوي واحد أما قيمة الرقم فتساوي سالب المكمل الثنائي للعدد أي لو أخذنا رقم مخزن في واحد بايت مثال = 11110110 بما أن البت السابعه=1 فأن الرقم سالب / نأخذ الآن المكمل الثنائي للعدد وهو 00001010 / القيمة تساوي -00001010 أي سالب عشرة .

وبتخزين الرقم بالأشارة يختلف النطاق الذي تأخذ كل وحده :

أسم الوحدة	المجال	
	من	الى
البايت	-128	+127
الكلمة	-32,768	+32,767
الكلمة المضاعفة	-2,147,483,648	+2,147,483,647
الكلمة الرباعية	-9,223,372,036,854,775,808	+9,223,372,036,854,775,807

تعليمات وأوامر الأسمبلي :

تتكون التعليمه الواحدة في الأسمبلي من تمثيل بسيط بالأحرف الأنجليزيه يقابله بالأرقام تعليمة لغة آله ، تتكون كل تعليمة من ممايلي : أولاً جزء الأمر وهو أمر يدل على نوع العملية المطلوبة مثل ADD (للجمع) ، الجزء الثاني هو الوسائط علماً بأن بعض التعليمات لايأخذ وسائط والجزء الآخر وسيطة واحدة فقط والبعض الآخر أكثر من ذلك ، تحدد هذه الوسائط الشئ الذي سيعمل عليه الأمر ،
فالأمر ADD لوحده عقيم لايدل على شئ لكن الأمر ADD AX,5 يدل على جمع الرقم 5 مع القيمة الموجودة في المسجل AX ويوضح المثال التالي بعض الأوامر

```
clc          ; بدون وسائط فقط أمر
dec ax       ; وسيطة واحدة فقط
mov cx,dx ; وسيطتين
```

لاحظ أن أي نص في شفرة الأسمبلي يأتي بعد الفاصلة المنقوطة هو مجرد تعليق

الوسائط ممكن تكون عدة أنواع :

1. معلومة فورية (مباشرة) (أي ثابته) مثال : 10 / 30 / "a"
2. مسجل (سوف يتم شرح المسجلات بالتفصيل في الدروس القادمه) مثال : / AX / EAX / BL
3. موقع ذاكره (يتم تحديده عن طريق العنوان) مثال : [200] / [[100] / [bx
4. متغير (وهو نفس السابق لكن بدل أن تحفظ أو تحسب العنوان يدوياً يقوم الأسمبلر بأستبدال المتغير برقم يدل على عنوانه) مثال : STR1 / INTVAL / VAR1 / count

مدخل الى الديبغر Debugge :

ها قد وصلنا الى واحد من أقوى البرامج المبيته في النظام فبواسطة الديبغ تستطيع عمل أشياء عجيبه وغريبه ، حسناً شغل الدوس وعند محث الأوامر أطبع debug ثم أنتر وستظهر لك علامه '-' دليل على أستعداد الديبغر على أستقبال أوامرك .

الأن دعنا نكتب هذا البرنامج الصغير

```
mov ax,2   ; نقل العدد 2 كمعلومة مباشرة الى المسجل أي-أكس
mov bx,3   ; نقل العدد 3 كمعلومة مباشرة الى المسجل بي-اكس
add ax,bx  ; أي-اكس / أي-اكس=أي-أكس + بي- جمع أي-اكي مع بي-اكس مع وضع الجواب في أكس
```

كيف تقوم بأذخال هذا الكود :

1. عند المحث '-' أدخل a100 أ ي أننا سنبدأ نكتب الكود من العنوا ن 100 ثم أضغط أنتر بالطبع
2. الآن أدخل كل تعليميه ثم أضغط أنتر ومع نهاية التعليمية الأخيرة أضغط أنتر مرتين

الأن قم بأدخال الرمز R ثم أنتر لترى حالة المسجلات
لاحظ أ ن المسجل AX يساو ي صفر وسترى أيضاَ ظهور التعليميه MOV ax,0002 وهي التعليميه التي عليها الدور في التنفيذ وليس المعلومة المنفذه ، الآن قم بطباعة الرمز T ثم أنتر لتنفيذ التعليميه التي عليها الدور هنا هي MOV AX,0002 سترى الآن أ ن المسجل AX أصبح يساو ي 2 وهذا مانتوقعه بالضبط وسترى أيضاً التعليميع التي عليها دور التنفيذ وهي MOV BX,0003 أدخل الرمز T ثم أنتر لتنفيذها لترى أ ن المسجل BX أصبح يساو ي 3 وسترى أيضاً التعليميه التي عليها الدور في التنفيذ وهي ADD AX,BX قم بأدخال الرمز T لتنفيذها ولاحظ كيف أ ن المسجل AX أصبحت قيمته مجموع العددين 3+2 وهو خمسة بينما بقى المسجل BX يساو ي 3 .

الآن بعدما عرفت كيف تكتب كود بسيط أخرج من الديبغر بالضغط على Q ثم أدخل مرة أخرى بكتابة الأمر Debug حتى تصفر المسجلات مرة أخرى أدخل التعليميه A100 ثم جرب تكتب كود من عندك ومع كل نهاية تعليميه أضغط أنتر وفي نهاية التعليميه الأخيرة أضغط أنتر مرتين
(ملاحظة لترى شفرتك بلغة الآله والأسمبلي أدخل الرمز U ثم أنتر مباشرة بعد إدخال الكود وقبل إدخال الرمز R)
أضغط R ثم أنتر لترى المسجلات قبل تنفيذ أ ي عملية ولترى التعليميه التي عليها الدور في التنفيذ أضغط T ثم أنتر لتنفذ التعليميه وترى النتائج والتعليميه التي بعدها وهكذا ولاتنسى أذا أرد ت أ ن تدخل كود جديد الخروج والعودة مرة أخرى الى الديبغر لتصفر المسجلات والذاكره
